Dungeon

Are you ready for an adventure?

Outline

1. Introduction
2. Requirements
   - Basic Functions
   - Optional Enhancement
   - Report
   - Video
3. Grading
4. Submission
5. Get Help

Introduction

In this homework, you will make a text-based RPG with the player exploring a dungeon and fight against unknown monsters.

You will implement it using C++ and put your OOP knowledge into practice. Header files and tutorial videos are provided. Try your best to accomplish it.

Let's go for an adventure! >>

Requirements

1. Basic Functions (70%)
   - Movement (10%)
   - Showing Status (5%)
   - Pick up Items (10%)
   - Fighting System (10%)
   - NPC (10%)
   - Game Logic (10%)
   - Hunger System (7.5%)
   - Room System (7.5%)
2. Optional Enhancement (10%)
3. Report (15%)
4. Video (15%)

Actions Menu

Allowing player to choose next action.

What do you want to do ?

A. Move
B. Check Status
C. Talk to Shop
D. Open Backpack

Movement(10%)

Allowing player to move from one room to the other.
e.g. Moving from Room No.1 to Room No.2, and then go back to Room No.1

Where do you want to go?

A. Go up
B. Go down
C. Go left
D. Go right

Showing Status(5%)

Dump out the player’ s information.

Status:
[Player’s Name]
> Health: 100/100
> Attack: 30
> Defense: 0
> Items: (if you have)
> (All you want to add…)

Pick up Items(10%)

Allowing user to pick up items and equip them onto the player.

Fighting System(10%)

Player can attack the monster, and the monster will either die or fight back if it is still alive. Moreover, the fight will loop until either one of them dies or the player retreats.

- Attack (kill the monster or the monster will fight back)
- Retreat (send player back to the previous room)

NPC(10%)

- Showing the script of NPC (what to say from the NPC)
- Trade with the player (player can get item(s) from the NPC)

Game Logic(10%)

Handle when the player win or lose.

Hunger System Design (7.5%)

Adding Hunger System Design.
Note that the UML does not include the design for a "Hunger System Design". You need to make up a design for coding by yourself.

The Hunger System Design should include three types of statuses.

- Hunger
- Thirst
- Poison

Add more statuses as you wish, and show your advanced design in Optional Enhancement of the report.

More information on the next page.

Hunger System Design (7.5%)

Hunger

- Must design a hunger status for holding how hungry the player is.
- The hunger status decreases as the player progresses. If the hunger status decreases to 0, the health of the player will start to decreases too.
- Must have at least three types of foods, gaining different amount of health and having different ways to acquire. e.g., from animals, from NPCs, from monsters.

Thirst

- Must design a thirst status for holding how thirsty the player is.
- The thirst status decreases as the player progresses. If the thirst status decreases to 0, the health of the player will start to decreases too, along with at least one debuff.
- Must have a way to acquire water. e.g., from lake, from NPCs, from environments.

Buff and Debuff means: https://smallgiantgames.helpshift.com/hc/en/4-empires-puzzles/faq/331-what-are-buffs-debuffs-status-effects/

Hunger System Design (7.5%)

Poison

- Must design a poison status for holding the poison the player has received.
- If the poison status has value, the health of the player will decrease the same amount each turn.
- Must have at least two different poison, having different value, duration time, and different ways to get. e.g., from getting hit by monsters, from food or water, from environments.
- Must have at least a way to acquire milk to detoxify. e.g., from NPCs, from animals.

Buff and Debuff means: https://smallgiantgames.helpshift.com/hc/en/4-empires-puzzles/faq/331-what-are-buffs-debuffs-status-effects/

Room System Design (7.5%)

Adding Room System Design.

Note that the UML does not include the design for a "Room System Design".

You need to make up a design for coding by yourself.

The Room System Design should include three types of rooms. Different rooms have different effects and consume different amounts of hunger and thirst.

- Desert
- Forest
- Swamp

Add more rooms as you wish, and show your advanced design in Optional Enhancement of the report.

More information on the next page.

Room System Design (7.5%)

Desert

- Thirst Decrease: Being in the desert causes the player's thirst status to decrease dramatically over time, in comparison to other environments.
- Sandstorms: Occasionally, sandstorms can occur in the desert, causing the player's hunger and thirst statuses dramatically to decrease over time until the sandstorm subsides.
- Oasis: Rarely, players may stumble upon an oasis in the desert, where they can replenish their thirst status.

Room System Design (7.5%)

Forest

- Hunger Consumption: Being in the forest causes the player's hunger status to decrease dramatically over time, in comparison to other environments.
- Wildlife: In the forest, players may encounter formidable predators like bears and tigers, posing a serious threat to survival.
- Lake: Within the forest, players may stumble upon a lake where they can replenish their thirst status.

Room System Design (7.5%)

Swamp

- Design by yourself.

Optional Enhancement(10%)

You are allowed to add additional functions, features or even visualize the gameplay to enhanced the game experience (as long as the system works).

- Please describe the additional features clearly in your demo video and especially in your paper report.
- Example
  - MP system
  - Cooling Down Time (CD)
  - Type counter (i.e. water counters fire, fire counters grass,
  - grass counters water)
  - DPs, tank, etc.
  - Record System(Save/Load the game)

Note

1. Our template is meant to speed up your development not to constraint your creativity. You can modify it or add extra functions whenever you want (as long as it still fulfill all the requirements).
e.g. You might change the return type of a function.

2. In both demo video and paper report, all types of rooms should exist in the map in order to demonstrate how you fit all the requirements.
e.g. Rooms of 1. monster, 2. NPC, 3. chest, 4. exit (boss).

3. You are allowed to design your own text-based game (complete different with our dungeon), but you must implement with OOP structure and describe in detail the OOP framework you are using in the report. (We will inspect your game more strictly than dungeon type.)

Grading

1. You can get the basic score (70%) after finishing all the basic requirements.
2. You might get a higher score (up to 110%) if you successfully enhance your dungeon game with other functions or abilities.
3. You must implement the assignment with inheritance and virtual functions, or there will be a 0 score. Please describe the structure of your inheritance and virtual functions clearly in the paper report.

Grading

4. There will be no demo section but a demo video and a detailed paper report instead.
5. Optionally enhanced functions or features should also be described clearly so we can easily get your ideas and know your efforts.
6. About video (15%):

- The video should be no more than 300 seconds .
- The video should demonstrate all your working functions and describe your UML design approach.
- Editing is needed, add some labels to emphasize your features. (But faking your results are not allowed. We will inspect your video strictly to follow your code.)

Grading

7. About report (15%):

The report should be clear and organized .

- The report should describe how you fulfill the requirements .
- The report must contain the following:
  - Implementation detailed
  - UML Design
  - Results
  - Discussion
  - Conclusion

8. No plagiarism! Any plagiarism will get 0 score.

Submission

1. A zip file named <HW1_studentID.zip>
   - A folder of all your codes and resources named <Dungeon_studentID>
   - A paper report named <report_studentID.pdf>
   - A demo video named <demo_studentID.mp4>
   - You will get 0 score if any file missed.

3. Any wrong submission format will be 5% penalty.
3. Uploaded onto E3 platform before 4/22 (Mon) 11:59 pm
4. Late penalty will be 10% per day .

Get Help

Tutorial

The playlist may help you.

https://www.youtube.com/playlist?list=PLolSxpCB-u1AkcGChEgFomFRhuLrFsOCu

Any question

- Lab time: Tuesday 6:30(p.m.) ~ 9:30(p.m.) EC315
- New e3 Forum: Others may have the same question.
- E-mail: cc to all TAs